APPLICATION NOTE

Marvell® ARMADA™ (PXA) 16x Plug Computer USB Webcam Streaming Internet Video Demo Setup

1. Purpose

This document describes how to install and customize the USB Webcam Network Video Streaming Demonstration for the Marvell® ARMADA™ (PXA) 16x Plug Computer. The package archive consists of a Linux 2.6.29 kernel compiled with built-in USB video-class device support, the video streamer/web server program, and a set of web pages that demonstrates how to display the video stream.

2. Requirements

The demo runs on the Marvell ARMADA (PXA) 16x Plug Computer and has been tested with versions 0.4, 0.5 and 0.6 of the root file system.

In theory, the demo should work with any USB web camera that complies with version 1.0 or later of the USB video class (UVC). Practical reality is such that some webcams purporting to be UVC devices do not work with the version of the Linux driver compiled into the kernel that is included with the demo package.

During development and testing of the demo, two such webcams were identified: a white-box model based on a controller chip from Aveo (VID:PID = 1871:0d01), and the Kodak Dual Webcam D101 VID:PID = 0ac8:3460). So having a few different candidates to try with the demo is prudent. These cameras have been used successfully:

- Kodak Family HD Webcam (VID:PID = 0603:8122)
- Logitech HD Webcam C510 (VID:PID = 046d:081d)
- Logitech QuickCam Fusion (VID:PID = 046d:08ca)
- Logitech QuickCam Orbit AF (VID:PID = 046d:0994)
- Logitech QuickCam UltraVision (VID:PID = 046d:08c9)

USB vendor ID (VID) and product ID (PID) are listed for the specific cameras tested with the demo. A given webcam model may encompass multiple hardware revisions, but each of these usually has a unique PID, thus permitting differentiation. Because this is a streaming video demonstration, the Plug Computer must be connected to a network along with the PC used for browsing. Although not required, a live Internet connection facilitates downloading the demo tarball directly onto the Plug.

3. Software and Licenses

This demonstration is built entirely from open source software. The packages used and their governing licenses are as follows:

Linux Kernel	www.kernel.org	GNU Public License version 2 (GPLv2)
Debian GNU/Linux	www.debian.org	Various
MJPG-streamer	mjpg-streamer.sf.net	GPLv2
Mollio CSS/HTML Templates	www.mollio.org	
GreyBox	orangoo.com/labs/GreyBox	GNU Lesser General Public License v2.1 (LGPL)
Cambozola	www.charliemouse.com/code/cambozola	GPLv2
Javascript rendering code	(by Richard Atterer in MJPG-streamer)	GPLv2

Copyright © May 2011 Marvell

4. Installation

1. Load the demo tarball onto the Plug Computer. The easiest way to do this, assuming the Plug has a live Internet connection, is to download it directly with `wget`. At a command prompt (see the *Marvell® ARMADA™ (PXA) 16x Plug Computer Development Kit User Manual* for options), enter…

```
wget http://url.of.demo
```

…where `url.of.demo` is the URL of the demo tarball (copied from the web browser and pasted into the terminal window for the sake of simplicity). After downloading the demo, install it as follows:

2. Extract the files from the tarball with the following command:

```
tar -xvjPf marvell-mjpg-streamer-demo.tar.bz2
```

The "P" option is necessary to install the script that starts and stops the demo in `/etc/init.d` and the symbolic link in `/etc/rc5.d` that launches the demo near the end of system boot. If the "P" option is omitted, the entire demonstration package is extracted to the working directory, and the demo will not launch automatically when the system is rebooted.

3. Mount the kernel partition. Use `/mnt` for the mount point as shown below if nothing is already mounted there or in a subdirectory below `/mnt`.

```
mount /dev/mmcblk0p1 /mnt
```

4. Unless otherwise modified, U-Boot automatically loads the zImage file to launch the boot sequence. Rename the existing kernel to back it up with, for example:

```
mv zImage previousZimage
```

5. Copy the kernel provided in the demo tarball to the kernel partition on the Plug microSDHC card:

```
cp /marvell-mjpg-streamer-demo/kernel/zImageDemo /mnt/zImage
```

6. Now unmount the kernel partition:

```
umount /mnt
```

7. The new kernel and demo are now installed. Before rebooting to launch the demo, take note of the Plug IP address:

```
ifconfig
```

The output should look something like this:

```
eth0      Link encap:Ethernet  HWaddr fa:d4:e0:00:32:b1
          inet addr:192.168.1.138  Bcast:192.168.1.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7940 errors:0 dropped:0 overruns:0 frame:0
```


Copyright © May 2011 Marvell

```
          TX packets:266 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:585947 (572.2 KiB)  TX bytes:50444 (49.2 KiB)
          Interrupt:22
```

8. Copy the Plug IP address, ensure the USB webcam is connected, and issue the reboot command:

```
reboot
```

9. When the Plug reboots, the demo start-up script is called, and the following status message is output:

```
MJPG Streamer Version.: 2.0
 i: Using V4L2 device.: /dev/video0
 i: Desired Resolution: 640 x 480
 i: Frames Per Second.: 15
 i: Format............: MJPEG
 o: www-folder-path...: ./www/
 o: HTTP TCP port.....: 80
 o: username:password.: disabled
 o: commands..........: enabled
```

If the webcam has a status LED that lights when active, it will be lit. When the demo is started with a webcam that is not compatible with the built-in USB video-class driver, an error occurs and displays in the status output from MJPG-streamer:

```
MJPG Streamer Version.: 2.0
 i: Using V4L2 device.: /dev/video0
 i: Desired Resolution: 640 x 480
 i: Frames Per Second.: 15
 i: Format............: MJPEG
Unable to set format: Invalid argument
 Init v4L2 failed !! exit fatal
 i: init_VideoIn failed
```

Unfortunately, there is no simple way to determine if MJPG-streamer fails to launch because of a given webcam's lack of compatibility with some aspect of the UVC specification, a shortcoming in the Linux 2.6.29 kernel UVC driver, or some problem with MJPG-streamer itself. The best option is simply to try another USB webcam.

5. Using the Demo

Point a web browser to the IP address noted in Step 1 above. The demo home page looks like that shown in Figure 1:

Copyright © May 2011 Marvell

Figure 1: ARMADA (PXA) 16x Plug Computer Webcam Demo Home Page

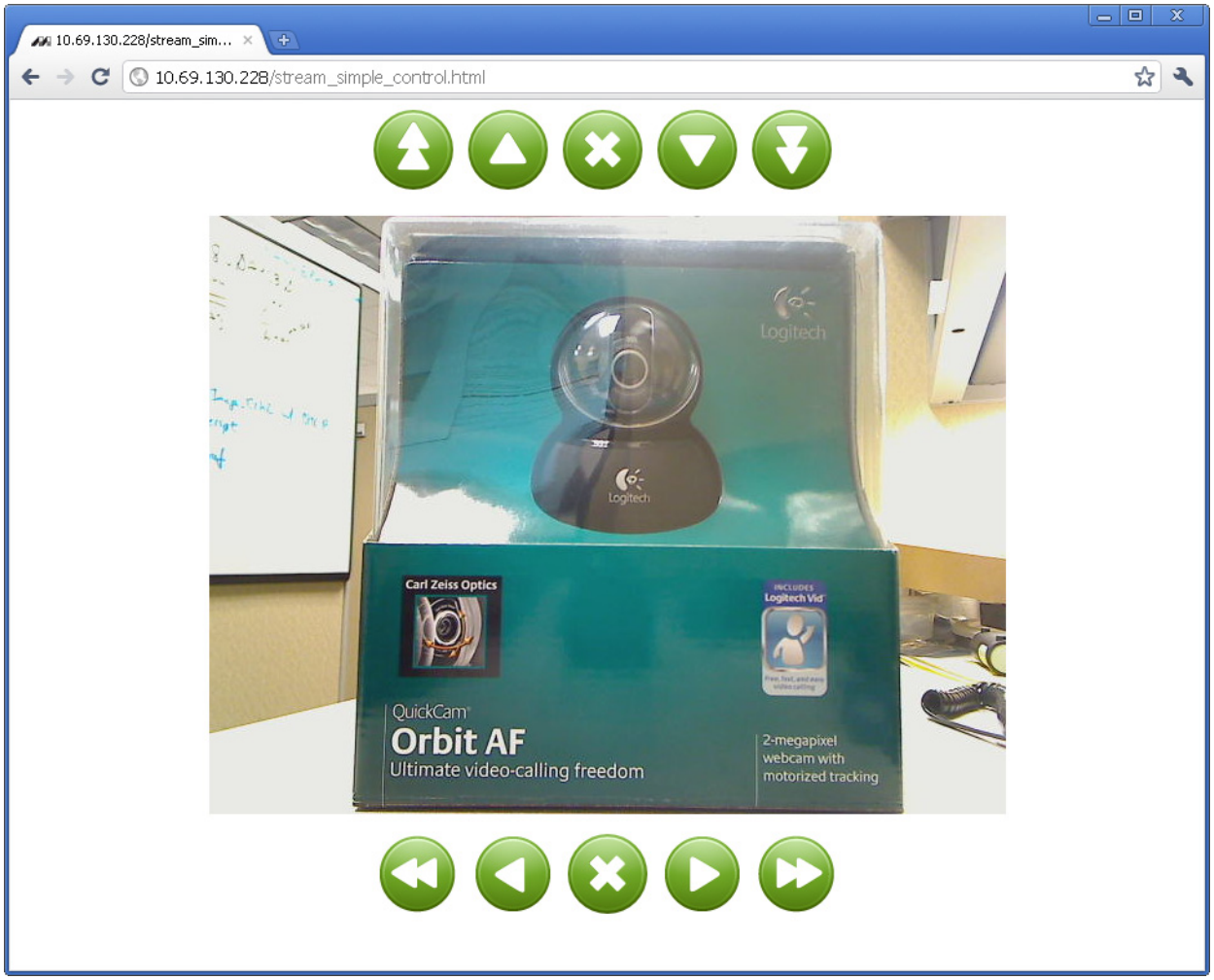
Marvell® ARMADA™ 168 P...
10.69.130.228/index.html#
ARMADA™ 168 Plug Computer Webcam Demo
About this Demo
Snapshot
Stream
Java
Javascript
MARVELL
Efficiently Serving Up Video with HTML and Linux
This demonstration uses the open source MJPG-streamer package running under Debian Linux on the ARMADA 168 Plug Computer to stream live video from a USB video class-compliant web camera for rendering in a browser.
MJPG-streamer has its own internal web server that is easily configured via command line switch to use any suitable TCP port. This allows it to run at the same time as a standard Linux-oriented web server like Apache, lighttpd, or thttpd.
Minimizing Overhead with USB Video Class Devices
Webcams that comply with the USB video class (UVC) are commonly available and preferred for embedded applications because they output Motion JPEG (M-JPEG) video. The camera controller IC performs the M-JPEG encoding in hardware, eliminating the need to do so in software on the embedded processor. The video stream can be transmitted to a web browser using HTTP, and unlike Flash or RTSP, no plug-in is needed.
MJPG-streamer at Work
The image displayed here was grabbed by the MJPG-streamer input plug-in. The HTTP request contains the GET parameters action=snapshot. This requests one single picture from the image input. To display another example, just click on the picture. To view the stream with any browser select Javascript or Java from the navigation bar at the top of the page.
Firefox and Google Chrome are able to render the M-JPEG stream directly. Click Stream to see this in action. Microsoft Internet Explorer 8 and earlier (version 9 was not tested) and Opera 11 cannot do this.
Learn more about the MJPG-streamer project here.
GLOBALSCALE
GPLUG-DISPLAY

Explanatory text on each web page covers the different ways in which the MJPEG video stream can be rendered in a browser. In particular, note the inability of Internet Explorer and Opera to render MJPEG. When using either of these browsers, the Java- or Javascript-rendering method must be generally used.

When the demo is launched with a Logitech QuickCam Orbit AF connected to the Plug, the simple example web pages include buttons to the control this webcam's motorized pan and tilt features (see Figure 2).

Copyright © May 2011 Marvell

Figure 2: Logitech QuickCam Orbit Web Page



The top and bottom rows of buttons in Figure 2 control the webcam's tilt and pan features, respectively. The double arrows perform a full-extent pan or tilt while the single arrows adjust the pan or tilt in five-degree increments. The "X" button in each row returns the pan or tilt to its center position. Note that all directions are relative to the camera, thus a user looking directly at the camera will see video to his/her right when panning left and vice versa.

6. Hacking the Demo

Modifying the demo is very easy. All of the web pages served up by MJPG-streamer's internal web server are located in the `/marvell-mjpg-streamer-demo/www` directory. The files ending with `_control.html` include the HTML `<form>` and `<input>` tags along with some associated Javascript used to send pan and tilt commands to the Logitech QuickCam Orbit AF.

The MJPG-streamer program is started by the `/marvell-mjpg-streamer-demo/marvell-mjpg-streamer-demo.sh` script. This line includes all of the parameters necessary to reconfigure the demo for a different frame rate or size:

```
./mjpg_streamer --input "input_uvc.so -f 15 -r 640x480" --output "output_http.so -p 80 -w ./www" &
```

Copyright © May 2011 Marvell

The input_uvc.so plug-in is the module that MJPG-streamer uses to acquire MJPEG video from a UVC-compliant webcam. The "-f" and "-r" options specify the frame rate and resolution, in this case, 15 fps and 640x480 pixels, respectively. The "-r" parameter can be an explicit frame size like 640x480 or a standard resolution term like QSIF, QCIF, CGA, QVGA, CIF, VGA, SVGA, XGA, or SXGA.

Smooth video streaming of almost any frame size is possible with proper frame rate and camera selection. For example, MJPG-streamer running on the ARMADA (PXA) 16x Plug Computer can transmit 1600x896 @ 5 fps video using the Logitech HD Webcam C510, even with the full Debian Linux desktop environment running. For reasons that are not entirely clear (perhaps certain camera controller ICs operate more efficiently), some USB webcams work better than others when transmitting high resolution or high frame rate video. Experiment with different camera models and vendors to see which work best.

Web pages and video are served up by MJPG-streamer's output_http.so plug-in. The TCP port used by the web server is specified with the "-p" option. This demo keeps things simple by running the web server on port 80, the standard port for HTTP traffic. It is entirely possible to run another web server alongside one or more instances of output_http.so provided each is assigned a unique TCP port. For instance, a single ARMADA (PXA) 16x Plug Computer can be connected to four webcams and run four instances of MJPG-streamer along with a secure, lightweight front end web server like lighttpd.

All of the HTML, CSS, GIF, PNG, and Javascript files that comprise the demo web site, along with the Cambozola Java video-rendering applet, are stored in the www subdirectory, which is passed to the output_http.so plug-in with the "-w" option. Note that MJPG-streamer can operate without a single page of HTML. To obtain a snapshot, specify the Plug Computer running MJPG-streamer as the source in an <img> tag as follows:

```
<img src="http://IP_ADDRESS_OR_HOSTNAME:PORT_NUMBER/?action=snapshot">
```

Valid attributes for <img> can also be placed in this tag as the tag is rendered by the web browser and has no impact on the operation of MJPG-streamer. So, if MJPG-streamer is running on port 8800 of the Plug Computer with IP address 10.69.130.228 and is configured to stream 640x480 frames, the <img> tag used to generate a 160x120 thumbnail for another web server is:

```
<img src="http://10.69.130.228:8800/?action=snapshot" style="width:160px; height=120px;">
```

Use ?action=stream in this tag to transmit video. As noted above, only Firefox and Google Chrome can continuously render a MJPEG stream. With Internet Explorer or Opera, video must be rendered with either Javascript code or the Cambozola Java applet. See the javascript.html and java.html files for these techniques. For Internet Explorer, the video presentation method can be made transparent to the end user by using conditional comments to insert IE-specific code when the HTML page is rendered. Consider this fragment:

```
<!--[if IE]>
  <applet code="com.charliemouse.cambozola.Viewer" archive="/cambozola.jar"
    <param name="url" value="/?action=stream" />
  </applet>
<![endif]-->
<![if !IE]>
  <img src="/?action=stream" />
<![endif]>
```

On Internet Explorer, the HTML engine ignores the <img> tag that refers to the stream action and instead loads the Cambozola Java applet to render video from MJPG-streamer. An option that is more complicated than Internet Explorer conditional comments but that handles a wider range of browsers is to interrogate the navigator.userAgent or navigator.product properties with Javascript that then modifies the HTML code identified by a specific of tag.

Copyright © May 2011 Marvell

Of course, the Javascript-rendering code provided in the MJPG-streamer sample files and used in this demo works with all modern browsers and can be used as a kind of least-common denominator. Unfortunately, even when interpreted by the fastest Javascript engines available, it still cannot achieve the kind of frame rates possible with browser-native MJPEG rendering or by the Cambozola Java applet.

Disclaimer

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of Marvell. Marvell retains the right to make changes to this document at any time, without notice. Marvell makes no warranty of any kind, expressed or implied, with regard to any information contained in this document, including, but not limited to, the implied warranties of merchantability or fitness for any particular purpose. Further, Marvell does not warrant the accuracy or completeness of the information, text, graphics, or other items contained within this document.

Marvell products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use Marvell products in these types of equipment or applications.

With respect to the products described herein, the user or recipient, in the absence of appropriate U.S. government authorization, agrees:

1) Not to re-export or release any such information consisting of technology, software or source code controlled for national security reasons by the U.S. Export Control Regulations ("EAR"), to a national of EAR Country Groups D:1 or E:2;

2) Not to export the direct product of such technology or such software, to EAR Country Groups D:1 or E:2, if such technology or software and direct products thereof are controlled for national security reasons by the EAR; and,

3) In the case of technology controlled for national security reasons under the EAR where the direct product of the technology is a complete plant or component of a plant, not to export to EAR Country Groups D:1 or E:2 the direct product of the plant or major component thereof, if such direct product is controlled for national security reasons by the EAR, or is subject to controls under the U.S. Munitions List ("USML").

At all times hereunder, the recipient of any such information agrees that they shall be deemed to have manually signed this document in connection with their receipt of any such information.

Copyright © 5/25/11. Marvell International Ltd. All rights reserved. Marvell, the Marvell logo, Moving Forward Faster, Alaska, Fastwriter, Datacom Systems on Silicon, Libertas, Link Street, NetGX, PHYAdvantage, Prestera, Raising The Technology Bar, The Technology Within, Virtual Cable Tester, and Yukon are registered trademarks of Marvell. Ants, AnyVoltage, Discovery, DSP Switcher, Feroceon, GalNet, GalTis, Horizon, Marvell Makes It All Possible, RADLAN, UniMAC, and VCT are trademarks of Marvell. All other trademarks are the property of their respective owners.

Intel XScale® is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Marvell Semiconductor, Inc.
5488 Marvell Lane
Santa Clara, CA 95054, USA
Tel: 1.408.222.2500
Fax: 1.408.752.9028
Email: commsales@marvell.com
www.marvell.com